SUMC

Multiprocessing Operating System (MPOS)

Requirements Document

Advanced Technology Laboratories
Government and Commercial Systems
Camden, New Jersey

SUMC MPOS

REQUIREMENTS

DOCUMENT

Submitted to:

George C. Marshall Space Flight Center
Marshall Space Flight Center
Huntsville, Alabama 35812

Referring to:

Contract NAS8-29072
Modification No. 2

Submitted by:

Advanced Technology Laboratories
Government & Commercial Systems Division
RCA Corporation
Camden, New Jersey 08102

Document Prepared by:

W. A. Helbig - RCA
L. C. Keller - M&S Computing
D. A. Nelson - RCA
T. T. Schansman - M&S Computing
L. E. Vandergriff - RCA

Approved by:

W. A. Clapp

W. A. Clapp
RCA Advanced Technology Laboratories

PREFACE

This document describes the requirements established for the Multi-Processor Operating System (MPOS) to be developed for a multi-processor configuration of the Space Ultrareliable Modular Computer (SUMC).

The first section of this document lists the major objectives of the system. The effect of the objectives on the requirements is summarized in Section 2. Section 3 then provides the requirements that will guide subsequent development.

TABLE OF CONTENTS

Section		Page
1.	SUMC/MPOS SOFTWARE OBJECTIVES	1
2.	OBJECTIVES/REQUIREMENTS CORRELATION	3
3.	SUMC/MPOS SOFTWARE REQUIREMENTS	5
	3.1 Job Management	5
	3.1.1 Job Origin	5
	3.1.2 Job Scheduling	5
	3.1.3 Job Scheduling Requests	5
	3.1.4 Resource Allocation	5
	3.1.5 Inter-Job Communication	6
	3.2 Task Management	7
	3.2.1 Task Scheduling	7
	3.2.2 Task Scheduling Requests	7
	3.2.3 Resource Allocation	8
	3.2.4 Inter-Task Communication	8
	3.3 Task Processing Support	10
	3.4 Input/Output Processing Support	11
	3.5 Fault Processing	12
	3.6 System Communication	13
	3.7 Testing/Debugging Support	14
	3.8 System Performance Monitoring	15
	3.9 System/360 Compatibility	16
	3.10 Other Language Support	17
	3.11 System Configurations	18

TABLE OF CONTENTS
(continued)

Section		Page
3.12	Storage Management	19
3.13	Operating System Maintenance	20
3.14	Job and Program Libraries	21
3.15	Management Support Utilities	22
3.16	System Design Concepts	23

1. SUMC/MPOS SOFTWARE OBJECTIVES

The basic requirements for MPOS and, therefore, its basic features, are derived from the objectives established to guide the system development.

These objectives are enumerated below:

Real-Time Oriented - MPOS must support the multiple real-time requirements of space vehicles, ranging from an unmanned vehicle to a large space station.

Batch Processing Support - MPOS must support multiple "batch-style" data processing applications running concurrently with the real-time applications. Where conflicts occur, however, this objective will be subordinated to Real-Time Oriented objective. These data processing applications include scientific calculations, and execution of support software (compilers, assemblers, link editors, etc.).

Multiprocessing - MPOS must enable the applications programmer to fully exploit the concurrent processing capabilities of the system. It must do this without requiring any special consideration by applications programmers, if they choose to consider MPOS a conventional multi-jobbing environment. MPOS must itself exploit the concurrent processing capabilities of the system, by allowing its functions to be used concurrently to the fullest extent possible. In addition, it must exploit the well defined, natural division of labor between SUMC CPU's and SUMC IOU's.

Virtual Storage - MPOS must exploit the potential advantages of simplified memory management and increased throughput without deteriorating the real-time performance of the system or the performance of configurations not requiring virtual storage.

Fault Tolerance - MPOS must support fault recovery as well as graceful degradation. MPOS must thus be able to isolate the effects of software/hardware faults to the component containing the fault. MPOS must further be able to correct the fault or remove the failing component from the system with as little user intervention as possible.

Reliability - To support unmanned missions, as well as to minimize required operator intervention and loss of time, MPOS must be as free of errors as current software technology allows and be "idiot-proof" against operator errors, application program errors, and hardware malfunctions. Structured programming methods offer the best hope for achieving reliable software: therefore they must be fully exploited in the development of MPOS.

Support Software Compatibility - To minimize support software development cost and programmer training cost, MPOS must first of all support a high degree of IBM S/360 compatibility. MPOS must execute application programs generated by IBM S/360

support software and maintain compatibility in program design conventions and facilities to the maximum extent possible. In addition, it is desirable that other support software currently under development by NASA for the SUMC can be used effectively.

Simple System Interfaces - MPOS must be able to support unattended system applications, therefore operator intervention should be an emergency measure rather than a standard operating procedure. For those applications (such as batch processing) where an operator is available, the system should be operable with a minimum of expertise. MPOS must be simple to learn, and use, in comparison with contemporary operating systems (e.g., OS/360, EXEC 8, MULTICS).

Similarly, the services MPOS provides to the application programmer should be simple to invoke and require a minimum amount of information from the programmer, if any. Internally, the functions of MPOS must be performed by highly decoupled, modular, programs. The system decomposition criteria of Parnas will be applied to achieve internal simplicity.

Adaptability - To support the wide range of potential missions efficiently, MPOS must be able to provide all functions necessary to support the most general application, as well as be able to contain only the minimum functions necessary to support more specialized applications. In addition, MPOS must allow the addition of application-dependent functions in specific areas for specific missions. Thus functional modularity must be exploited to enable the system generation of tailored versions of MPOS which contain only those operating system components needed to support a given mission.

System Visibility - The performance behavior of the system is of high importance in most real-time systems. The complexity of the system is such that without specific provisions to provide the user with the necessary feedback, it will be most difficult to understand what is going on in the system. At a minimum, MPOS must therefore support measurement tools and provide simple methods for tuning the system. Repeatability and reproducibility must be readily achievable in test environments.

2. OBJECTIVES/REQUIREMENTS CORRELATION

The requirements are grouped into relatively standard functional areas. The effect of the objectives on the requirements is quite complex. That is, multiple objectives may influence a single requirement and, conversely, a single objective may affect multiple requirements across multiple groupings.

To guide the reader in the evaluation of the effect that the objectives have on the requirements, a correlation matrix is presented in Figure 2-1.

It should be recognized that the requirements are selected based on current operating systems technology. In addition, the requirements so selected are one set out of many that can meet the stated objectives. During the functional design, other better or preferred requirements may be identified. Thus the requirements listed in the following sections are anticipated to be subject to change as development continues.

OBJECTIVES/REQUIREMENTS CORRELATION

	Job Management	Task Management	Task Processing Support	Input/Output Processing Support	Fault Processing	System Communication	Test/Debug Support	Performance Monitoring	S/360 Compatibility	Other Support Software	System Configuration	Storage Management	O/S Generation & Maintenance	Job/Program Libraries	Management Support Utilities	System Design Concepts
Real Time Oriented	X	X	X	X		X	X	X		X		X			X	X
Batch Proc. Support	X	X	X	X		X			X	X		X				
Multiproc.	X	X	X		X	X					X					X
Virtual Storage	X											X				
Fault Tolerance				X	X	X					X	X				X
Reliability					X											X
Support Software Compatibility									X	X						
Simple System Interfaces				X	X	X	X					X				X
Adaptability						X					X		X			X
System Visibility								X							X	

Figure 2-1.

3. SUMC/MPOS SOFTWARE REQUIREMENTS

3.1 Job Management

- The system must provide support for the concurrent control of multiple real-time and batch processing type jobs.

3.1.1 Job Origin

- Jobs can be entered into the system either from an on-line job library or via system input devices.

3.1.2 Job Scheduling

- Job scheduling is the allocation of sufficient job-level resources to ready a job for execution of its tasks (reference Task Management).
- Jobs are scheduled based on the availability of resources and in accordance with assigned priorities and/or deadlines.
- Before a job is scheduled, the system shall be able to adjust the job priority upwards as directed by manual intervention or in accordance with a job deadline algorithm.
- A job, once scheduled, will normally remain scheduled until its completion.
- For a certain class (override) of high-priority jobs, the system shall provide for cancelling of low-priority jobs as required to free resources necessary for execution of the override-priority jobs.

3.1.3 Job Scheduling Requests

- Job scheduling requests may be issued at any time by a system operator, by another job, or by means of a pre-defined time/event based schedule.

3.1.4 Resource Allocation

- A minimum number of resources shall be allocated on the job level.

- Job level resources shall be allocated by the system to prevent resource conflicts and reduce the number of times the active jobs must contend for resources.

- The system shall perform automatic resource allocation with minimum resource specification information supplied by the user.

3.1.5 Inter-Job Communication

- Jobs must be able to communicate with each other via common data pools under system supervision to insure data integrity.

3.2 Task Management

- The system must allow the job to be subdivided in units that can be executed sequentially or concurrently on multiple processors or in a multi-programming mode on a single processor. These units are called tasks.

3.2.1 Task Scheduling

- Task scheduling is the allocation of sufficient task-level resources to allow a task to start execution.
- Tasks are scheduled based on the availability of resources and in accordance with a combination of job and task priorities.
- Provisions shall be made to allow the user to modify or replace the standard scheduling algorithm at system generation time. But the standard scheduling algorithm must be sufficiently parameter driven to allow flexibility in task scheduling.
- Tasks shall be involuntarily suspended (pre-empted) when CPU allocation is required for execution of a higher priority task.
- The system shall permit tasks to voluntarily suspend themselves pending recognition of one or more time and/or event criteria (reference Task Scheduling Requests).

3.2.2 Task Scheduling Requests

- Task scheduling requests can be initiated dynamically by tasks within a job or based on pre-defined information at the time the job is scheduled.
- Task scheduling requests shall be dealt with as a function of the following criteria specified with the scheduling request.

Immediate

- The system shall accept requests for the immediate scheduling of a task.

Time

- The system shall provide for the scheduling of a task at a specified absolute time.

- The system shall allow a task to be scheduled at a time relative to the time of the request.

- The system shall allow the scheduling of a task at a fixed, periodic rate.

Event

- The system shall permit tasks to be scheduled upon the occurrence of one or more events. Events are such as the following:

  - External attention requests,

  - Error conditions, and

  - Inter-program flags.

Time/Event

- Combinations of time and event scheduling shall be provided:

  - Scheduling at a specified time interval following the occurrence of one or more events.

  - Scheduling upon the occurrence of one or more events after a specified time.

3.2.3 Resource Allocation

- Resources shall be allocated dynamically during task execution to the extent possible.

- The system shall perform automatic resource allocation with minimum resource specification information supplied by the user.

3.2.4 Inter-Task Communication

- The system shall provide common data facilities for inter-task communication with adequate controls for insuring integrity of the data.

- The system shall also allow tasks to communicate via parameter lists with data passed by value.

- The system shall provide facilities for signalling occurrence of events between tasks.

3.3 Task Processing Support

- The system shall provide the current real-time to application tasks upon request.
- The system shall provide interval timer services to application tasks using one or more interval timers with a fixed base.
- Tasks shall be able to request allocation of temporary working storage (dynamic main storage allocation).
- MPOS will support supervised sharing of common subroutines among application tasks.

3.4 Input/Output Processing Support

- The system shall provide support for application task communication with a variety of I/O devices including both real-time devices and standard record-oriented peripheral devices.

- MPOS shall support the Data Management system hardware by performing resource allocation and failure recovery to the extent required.

- MPOS shall relieve application tasks of concern for physical device characteristics such as data format and device address.

- MPOS shall support both synchronous I/O (task execution is suspended until the requested I/O operation is completed) and asynchronous I/O (task execution is not suspended).

- The system shall automatically attempt to recover from I/O errors detected by the hardware and, only upon recovery failure, notify the requesting task of the error.

3.5 Fault Processing

- MPOS shall insure that the effect of faults (hardware or software) is limited to the task in which the error occurred where possible.

- Recovery of faults will be transparent to the application task or systems operator to the fullest extent possible.

- Failed hardware components must be identified and isolated logically and physically (where possible) from the remainder of the system.

- Upon detection of software faults by the system, the failed software components shall be allowed to pass control to their own fault handling procedures.

- MPOS shall make provisions to log the faults encountered and the associated action taken.

3.6 System Communication

- System initialization shall allow utilization of automated procedures with operator override of adjustable system control parameters.

- MPOS shall provide operator interface facilities to enable an operator to perform the following functions:

  - Load jobs into the system

  - Control the scheduling of jobs

  - Control of the Data Management System configuration

  - Request status of Data Management system resources

  - Request job status information

- MPOS shall communicate to the operator the occurrence of certain critical events such as automatic Data Management system reconfiguration.

- The system shall be as unobtrusive as possible and require minimal operator intervention. The system will not rely on operator intervention and will provide alternatives in the event that operator responses are not provided.

3.7 Testing/Debugging Support

- MPOS shall support facilities for program testing/debugging which neither affect normal system operation nor require information other than that supplied by the standard support software.

- Interfaces shall be available for requesting snapshot and/or post-mortem storage dumps for various types of information in a variety of formats.

- System services shall be provided for tracing application task execution as directed by the user.

- A special interactive test mode shall be supported to permit on-line checkout of application tasks.

- Support of test mode simulation of event occurrences will be provided.

3.8 System Performance Monitoring

- MPOS shall support the capability to monitor various system performance parameters and maintain a log of such data for off-line analysis.

- MPOS shall log all hardware errors to the extent possible and software errors detected in operational real-time tasks.

- The system shall be capable of maintaining a summary of system service requests.

3.9 System/360 Compatibility

- The system shall be able to use S/360 support software to develop application programs for the SUMC/MP subject to specified ground rules and limitations for the following languages:
  - Assembler
  - FORTRAN IV Compiler
  - PL/I Compiler
- For user program interfaces, MPOS shall utilize concepts and provide services functionally compatible with OS/360 control program and job control services where feasible.

3.10 Other Language Support

- MPOS must be able to support GOAL.
- It is desirable that MPOS interface with HAL.
- It is desirable that MPOS interface with object programs produced by the NASA MSFC Computation Center support software.

3.11 System Configurations

- MPOS shall be able to support the maximum possible configuration as well as the minimum possible configuration (1 CPU, 1 IOP) without requiring system generation to adjust to configuration changes (configurations to be defined).

- MPOS shall be able to allow on-line partitioning of the hardware into logically independent computer systems.

3.12 Storage Management

- Storage shall be managed in a manner as transparent to the application software as possible.
- MPOS shall include facilities to monitor thrashing and take corrective action.
- In general, a pre-paging algorithm will be used to pre-page part (working-set) or all of a task prior to starting active execution.
- It must be possible to lock pages into physical memory until explicitly released.
- Demand paging shall be available for use if no pre-paging is specified or insufficient physical storage is available to perform all pre-paging requirements.

3.13 Operating System Maintenance

- System generation facilities capable of tailoring MPOS to meet specific requirements shall be provided to permit the user to specify the Data Management system hardware configuration, select optional software features, and include modules provided by the user.

- System generation input specifications shall be validated to guarantee the integrity of the system produced.

- Facilities shall be provided for updating individual system modules without the need for re-generating the entire system.

3.14 Job and Program Libraries

- The system shall provide support for maintaining libraries of program load modules.
- Facilities shall be provided for combining individual object programs into load modules.
- The system shall provide facilities for maintaining catalogued procedures to describe the structure of and dependencies between the various jobs which enter the system.
- The system shall provide facilities for maintaining program structure tables on a per task basis.

3.15 Management Support Utilities

- Test programs shall be provided which exercise and verify operation of all system functions.

- Data reduction facilities shall be provided for the analysis of information gathered by MPOS in the error, usage, and performance logs.

3.16 System Design Concepts

- MPOS shall be partitioned into layers (e.g., levels of abstraction) to the extent feasible.
- Structured programming principles shall be employed for each development phase to which they are applicable.
- MPOS shall employ functional modularity to insure that it can be closely tailored to the actual missions at system generation time.
- MPOS must be simple in comparison with contemporary operating systems (e.g., OS/360, EXEC 8, MULTICS) to enhance system reliability and maintainability.
- All information crossing interfaces into MPOS must be fully validated to make MPOS "idiot-proof".
- MPOS execution must not depend on specific modules being available, but must be able to dynamically adapt to any valid configuration (configurations to be defined).
- Concurrent execution of MPOS by multiple modules must be performed to the fullest extent possible.